

Agents for Cyber Defense

DATE

September 21, 2026

INTRODUCTION

AI agents hold significant promise for strengthening cyber defense. As frontier models have become more capable, agents powered by the most advanced general-purpose AI can help defenders protect critical networks, systems, and data at far greater scale and with much greater speed. By coupling advanced reasoning with persistent memory and tool use, agents can enable defenders to accelerate the detection and remediation of vulnerabilities, identify and counter threats, and assist security operations teams. Although attackers can leverage agents of their own in offensive cyber operations, the use of agents in cyber defense has the potential to bolster defenders as well, with significant implications for the security and resilience of the broader digital ecosystem.

This publication outlines how AI agents can assist defenders across the software development lifecycle, during development, in pre-release audits, and on deployed systems. Drawing on input from security experts across the member firms of the Frontier Model Forum (FMF), the issue brief builds on a prior publication on [AI for Cyber Defense](#) from 2024. Reflecting the increase in agentic capabilities, the issue brief focuses on five core use cases of agents for defensive operations: simulating adversarial behavior, aiding threat intelligence research and analysis, augmenting security operations, detecting and patching vulnerabilities, and improving code security. The brief closes with considerations for effectively deploying agents for cyber defense.

AGENTIC CAPABILITIES AND SECURITY TOOLING

Agents combine the underlying reasoning capabilities of frontier models with persistent memory, tool use, and the ability to iterate and

validate results. Compared with large language models, which are well-suited to discrete, process-specific cybersecurity tasks, agents are particularly useful in workflows that demand iteration, validation, tool use, and parallel task execution. They can increase efficiency by handling repetitive, high-volume work that would otherwise consume analyst time, such as triaging the hundreds to thousands of alerts a security team receives each day.

An agent's effectiveness depends heavily on the tools it can access, the harness^[1] built around the model, and the integration of the agent system into its operating environment.^[2] Equipping agents with specialized security tools, such as [Ghidra](#), [IKOS](#), or [OWASP ZAP](#), can meaningfully enhance their ability to perform malware analysis, detect vulnerable code, or identify threats. These tools allow agents to interact directly with code, binaries, and security infrastructure to complete cybersecurity tasks in ways that LLMs alone cannot.

Defensive Use Cases

AI agents provide several important uses for cyber defense. Although the use cases below are not exhaustive, they span the defensive lifecycle. Agents can stand in for adversaries during security testing, speed up threat intelligence and malware analysis, ease the alert burden inside security operations centers, and find, fix, and harden vulnerable code. From probing systems before attackers can exploit vulnerabilities to triaging threats once they are flagged, the use cases reflect where agents already have the capability to be useful to defenders.

Simulating Adversarial Behavior

Security testing, which involves probing an organization's defenses from the perspective of an attacker, is a highly skilled, time-consuming and resource-intensive process. Security testing often involves redteaming, which simulates threat actors carrying out real-world attacks on an organization, and penetration testing (pentesting), a focused evaluation of specific systems, applications, or network segments for vulnerabilities. Both are critical for identifying security gaps, such as misconfigured systems, default or reused credentials, and exposed services, and for assessing an organization's detection and response capabilities.

Penetration testers rely on a variety of tools to complete pentesting tasks, such as network scanning and fingerprinting, resource enumeration, vulnerability scanning, exploitation, and post-exploitation.^[3] By calling tools like Nmap for network scanning or Metasploit to exploit vulnerabilities, pentesting agents can simulate real attacks against running applications.^[4] One purpose-built agent scaffold outperformed nine out of ten human professionals at pentesting, and agent-based redteaming can compromise critical assets across realistic multi-host enterprise networks.^[5] Working under the direction of human testers, AI agents can automate parts of security testing processes, freeing skilled testers to focus on more challenging problems, such as chaining findings into realistic attack paths and assessing which findings

represent real business risk.

Agents can also support red/blue team testing, which pairs agents that simulate adversaries' techniques with agents that validate detection coverage and generate new or improved defenses when novel techniques surface.^[6] Running this loop in an execution environment that mirrors an organization's real codebases and production systems grounds the results. Agents execute commands against test systems and validate detections against real log data, so every claimed technique and every proposed detection rule is backed by observable evidence.

Threat Intelligence Research and Analysis

Security practitioners often face a barrage of security-related information requiring them to pull out valuable insights from threat reports, Open-Source Intelligence articles, and manually correlate indicators of compromise (IOCs). Agentic systems, including specialized sub-agents equipped with a set of tools, can be actioned to support threat analysts on specific tasks.^[7] For example, a question about a malicious file can route to a malware analyst agent that leverages Ghidra to analyze compiled software binaries. Reverse engineering these machine-readable files is essential for malware analysis because threat actors rarely distribute human-readable source code, and analysts must dissect these raw binaries to understand what the malware does and how to stop it.

AI agents can also translate raw threat intelligence into production-ready detection rules, the logic that flags suspicious attacker or user behaviors and triggers an alert.^[8] Developing and validating these rules can take hours to days. Post-deployment, detection rules require ongoing management, including assessing whether they remain effective and updating rules as new threats emerge. Agents can meaningfully accelerate detection development by supporting humans to write, validate, and maintain detection rules.^[9]

Augmenting Security Operations Workflows

Security Operations Centers (SOCs), the teams responsible for monitoring and responding to threats across their infrastructure, are often overwhelmed by the volume of security alerts and threat intelligence signals. Enterprise SOCs can receive hundreds to thousands of alerts per day with an average investigation time of 30 to 70 minutes per alert. Log files, information about the operations, activities, and usage patterns of an application, server, or system used to detect and investigate cybersecurity incidents, can contain millions of entries that need to be parsed quickly and reliably in response to a threat.

Agents can minimize the operational overhead of responding to and prioritizing security alerts, but their utility to SOC personnel can vary by task.^[10] SOC workflows involve standardized, repeatable procedures used to detect, triage, investigate, and remediate cyber threats. SOC workflows involve organizing analysts into three tiers:

- Tier 1 handles the highest volume of work at the lowest complexity.
- Tier 2 investigates escalated incidents requiring deeper analysis.
- Tier 3 tackles the most complex threats and proactive research.

Tiers 1 and 2 are well suited to operational uplift from agent assistance.^[11] SOC automation, log analysis, and threat triage represent immediate opportunities where agents can provide meaningful support. Because SOC workflows involve large amounts of information, AI proves particularly valuable at extracting the important details from raw data. Agents can assist with offline batch analysis, the process of analyzing historical data at scheduled intervals rather than in real-time, and retroactive analysis to automate detection, as well as automating log analysis that ultimately leads to threat identification.

Agents can be deployed across multiple SOC functions to maximize this potential. Agents can specialize in alert triage to quickly sort through incoming signals, while others assist with writing queries to identify suspicious patterns.^[12] Additionally, agents can take newly disclosed [Common Vulnerabilities and Exposures](#) and automatically create detection rules, enabling rapid response to emerging threats.

Detecting and Patching Vulnerabilities

Detecting vulnerabilities in code is one of the most promising areas of agent deployment in cyber defense.^[13] As a result, agents are significantly accelerating AI-powered vulnerability research. Frontier AI companies are building agents to find vulnerabilities at scale through fuzzing^[14] (bombarding programs with vast amounts of input data to expose memory corruption, integer overflow, and parsing errors), variant analysis (systematically searching for bugs similar to previously fixed vulnerabilities), and code repository scanning.^[15] In some cases, agents are embedded directly into the development pipeline, connecting to GitHub repositories to analyze code and flag likely vulnerabilities based on threat models built from repository and commit history.^[16]

AI agents equipped with security analysis tools, including static analysis, dynamic analysis, differential testing, fuzzing, and SMT solvers, can scrutinize code patterns, control flow, and dataflow to identify root causes of security flaws and architectural weaknesses in code early in the software development lifecycle.^[17] By integrating code review into existing workflows, agents can help identify and automatically fix vulnerabilities before code reaches production.^[18]

Detecting vulnerabilities, however, is only a first step toward improving security.^[19] To reap the most benefits from vulnerability discovery, organizations must be able to prioritize which vulnerabilities to address and develop patches suitable for deployment in a timely manner. As the number of AI-reported vulnerabilities increases, triage becomes the bottleneck. This constraint is particularly acute for critical infrastructure operators, whose operational technology (OT) environments often cannot be patched on demand.

Fixing a vulnerability involves analyzing the code, pinpointing the root cause of the

vulnerability, patching the vulnerable code, and verifying that the fix is accurate and does not degrade performance. Agents may help by analyzing error information from fuzzing and generating code fixes or identifying root causes of security flaws with access to tools like debuggers, code interpreters, and source code browsers.^[20] Agents can then verify fixes through automated testing before human review.

Identifying vulnerabilities and developing patches are critical, but patch deployment remains a persistent challenge.^[21] Applying a patch means confirming it doesn't break existing functionality, testing compatibility with current hardware, operating systems, or applications, and coordinating the system downtime required to update software, firmware, or operating systems. Even after patches become available, it can take weeks or months for organizations, particularly under-resourced organizations, to apply these updates.^[22]

Improving Code Security

Finding and fixing vulnerabilities addresses flaws in code that has already shipped. The furthest-upstream defense is to prevent vulnerabilities from being introduced in the first place. As frontier models generate increasingly large amounts of code, there's a corresponding need to validate code security and ensure AI-generated code is secure-by-design.^[23] While AI models can produce functionally correct code, they can also introduce security vulnerabilities like unchecked memory allocation and missing input validation.^[24] Beyond validation of existing codebases, other applications are emerging to secure code out of the box using secure frameworks that avoid common issues, such as weak cryptography, configuration-dependent bugs, Cross-Site Scripting, and code injection.^[25] Code hardening involves proactively looking at a codebase for issues like memory safety exceptions and modifying code to ensure all buffers are protected before a vulnerability can occur.^[26]

Many vulnerabilities in code can be traced back to software written in languages such as C and C++, which are vulnerable to exploits that target how computer memory is handled.^[27] Addressing these memory safety vulnerabilities could have an especially broad impact, because they can potentially be eliminated wholesale by translating code into memory-safe languages like Rust. In the near term, some agents can proactively rewrite existing code to use more secure data structures and APIs. However, these large-scale refactoring projects remain difficult to operationalize at scale, and memory safety is only a small part of the application security picture.

CONSIDERATIONS FOR DEPLOYING CYBER DEFENSE AGENTS

Deploying AI agents in cyber defense is not simply a matter of acquiring more capable models. The value an agent delivers depends on the maturity of the environment it enters, the use cases it is pointed at, the scaffolding and safeguards around its own operation, and the organization's ability to measure whether the agent is working as intended.

The following considerations frame where agents help, where they fall short, and

what defenders should put in place before granting them greater autonomy.

- **Get the Fundamentals Right.** Offensive cyber operations are often opportunistic, targeting unpatched or misconfigured systems, overly broad permissions, and weak credentials. Attackers rarely need a sophisticated exploit to compromise a target. The most effective way to improve an organization's security posture remains applying fundamental cybersecurity hygiene.^[28] Agentic code analysis and vulnerability discovery deliver far less value when these foundations are missing. Before investing in autonomous agents, organizations should ensure that fundamentals, including strong authentication, timely patching, and disciplined vulnerability management, are already in place.
- **Select Appropriate Use Cases.** A central challenge is ensuring that AI security tools surface high-impact findings and true positives. False positives are costly. They consume analysts' time, erode trust in automated systems, and can lead to alert fatigue that causes teams to overlook genuine threats. For this reason, preliminary applications of frontier AI in cybersecurity tend to be cautious, focused on automating and improving existing human-led security workflows before deploying more autonomous systems. Effective deployment also depends on high-quality datasets, such as realistic attack data, which are needed both to train agents and to validate their results. Defenders should select use cases for which they have reliable domain-specific training data to customize agents to a specific cybersecurity task.
- **Deploy in Stages and Match Autonomy to Risk.** As organizations build confidence in AI-powered security tools, deployment strategies evolve. Deployment is best approached as a staged process in which organizations first automate immediate operational bottlenecks, alert triage, log analysis, detection-rule drafting, before pursuing more transformative applications. The degree of autonomy granted to an agent should match the context and the consequences of error. Teams that successfully integrate AI into existing workflows often progress to granting agents greater autonomy, particularly in lower-risk areas like vulnerability scanning or code analysis where false positives may carry limited operational impact.
- **Secure the Agents Themselves.** As defenders delegate responsibility to AI agents, those agents become high-value targets. Subverting an agent that guards a network could give an attacker broad, undetected access. This makes reliability and adversarial robustness decisive for deployment in high-security contexts. Before agents can be trusted with greater autonomy, they must be [hardened](#) against prompt injection, data poisoning, and evasion attacks. Until the field of agent security matures, a zero-trust posture, in which agents are treated as untrusted components and their actions are constrained, logged, and verified, is a prudent default.^[29]

- Ensure Meaningful Oversight.** Real-time agent monitoring allows organizations to flag unsafe agent actions as they happen.^[30] Several complementary approaches can strengthen oversight. [Chain-of-Thought monitoring](#) examines an agent's reasoning traces for signs of unsafe behavior, such as reward hacking, alignment faking, and scheming.^[31] Synchronous monitoring involves separate LLMs reviewing each of the agent's proposed actions before they are executed.^[32] Human oversight also remains important, particularly in sensitive or mission-critical environments. In practice, this means escalation and approval workflows that route consequential decisions to a person, and tight integration with existing SOC tooling and playbooks so that agents work within established processes rather than around them. As agents handle more of the routine execution, defenders move into a supervisory role, setting direction, checking the agents' work, and deciding when to intervene. The oversight practices and standards that organizations build now will largely determine how much autonomy these systems can safely be given as they grow more capable.
- Invest in Better Defensive Evaluations.** Evaluation is one of the most significant blockers to wider deployment. Earning the trust of the defenders who rely on these agents depends on consistent performance under realistic operating conditions. Many existing cyber evaluations are inherited from offensive security.^[33] These evaluations can measure whether an agent succeeds at a discrete attack task but reveal little about defensive judgment, and they rarely test agents against the robust, adversary-driven conditions defenders actually face. A new wave of defense-focused evaluations is beginning to close this gap, including CyberSOCEval, ExCyTIn-Bench, and ZeroDayBench.^[34] However, the field still needs greater investment in defensive evaluations that assess an agent's overall ability to make sound decisions under uncertainty, the same standard applied to human analysts.^[35] Just as important, these evaluations should test defensive performance while adversaries are actively attacking.

CONCLUSION

Agents can reduce the burden to defenders across the defensive lifecycle. Defensive agents are already simulating adversaries, accelerating threat intelligence, triaging security alerts, and finding and fixing vulnerable code. While these capabilities enhance cyber defenders, realizing their potential also depends on the choices surrounding deployment. Agents introduce new risks of their own, since a defensive agent is also a high-value target. For defenders, this means getting the cybersecurity fundamentals right first, securing the agents against attack, ensuring oversight, and building evaluations that measure defensive judgment under realistic adversarial pressure.

Autonomous cyber defense is still an emerging practice and its operational norms remain nascent. With sufficient testing, validation, and oversight, AI agents used as

force multipliers under human direction have the potential to shift the balance toward defense, strengthening the security and resilience of the digital ecosystem. The FMF welcomes continued engagement from defenders and cybersecurity researchers, and will revisit emerging practices and norms for cyber defense as frontier cyber capabilities advance.

FOOTNOTES

1. The software infrastructure surrounding an AI model. It manages everything (tool execution, memory, context management, state persistence) except the model reasoning.
2. Justin W. Lin, et al., "[Comparing AI Agents to Cybersecurity Professionals in Real-World Penetration Testing](#)," arXiv, March 3, 2026.
3. Zachary Ezetta and Wu-chang Feng, "[PentestMCP: A Toolkit for Agentic Penetration Testing](#)," arXiv, October 4, 2025; Luca Gioacchini et al., "[AutoPenBench: Benchmarking Generative Agents for Penetration Testing](#)," arXiv, October 28, 2024; XBow, "[Threat Hunting with AI: Why it Matters and the Role of Autonomous Pentesting Tools](#)," August 24, 2026.
4. Xiangmin Shen et al., "[PentestAgent: Incorporating LLM Agents to Automated Penetration Testing](#)," ASIA CCS '25: Proceedings of the 20th ACM Asia Conference on Computer and Communications Security, August 24, 2025.
5. Justin W. Lin, et al., "[Comparing AI Agents to Cybersecurity Professionals in Real-World Penetration Testing](#)," arXiv, March 3, 2026; Brian Singer et al., "[Incalmo: An Autonomous LLM-assisted System for Red Teaming Multi-Host Networks](#)," arXiv, November 22, 2025.
6. Daniel Weiss, "[How Amazon uses AI agents to anticipate and counter cyber threats](#)," Amazon Science, November 24, 2025.
7. Google, "[Actionable threat intelligence at Google scale](#)," 2026.
8. Aditya Kumar, "[How AI Agents Are Turning Threat Intelligence Into Validated Detections](#)," Azure Infrastructure Blog, April 23, 2026.
9. Fatih Bulut et al., "[AVDA: Autonomous Vibe Detection Authoring for Cybersecurity](#)," arXiv, March 30, 2026.
10. Prophet AI, "[State of AI in Security Operations 2026](#)," 2026.
11. Payal Chakravarty and Vijay Ganti, "[The dawn of agentic AI in security operations](#)," Google Cloud, April 28, 2025; Lauren Deason et al., "[CyberSOCEval: Benchmarking LLMs Capabilities for Malware Analysis and Threat Intelligence Reasoning](#)," arXiv, September 15, 2025.
12. Payal Chakravarty and Vijay Ganti, "[The dawn of agentic AI in security operations](#)," Google Cloud, April 28, 2025.

13. DARPA, "[AI Cyber Challenge marks pivotal inflection point for cyber defense](#)," August 8, 2025.
14. Fuzzing is a frequently employed tactic to expose security and reliability issues.
15. TJ Byun et al., "[Introducing AutoPatchBench: A Benchmark for AI-Powered Security Fixes](#)," Meta, April 29, 2025; Google, "[From Naptime to Big Sleep: Using Large Language Models To Catch Vulnerabilities In Real-World Code](#)," November 1, 2024.
16. OpenAI, "[Codex Security: now in research preview](#)," March 6, 2026.
17. Raluca Ada Popa and Four Flynn, "[Introducing CodeMender: an AI agent for code security](#)," Google DeepMind, October 6, 2025.
18. Anthropic, "[Automate security reviews with Claude Code](#)," August 6, 2025.
19. Ruth C., "[10 questions to ask when using AI models to find vulnerabilities](#)," National Cyber Security Centre, May 11, 2026.
20. Jan Keller and Jan Nowakowski, "[AI-powered patching: the future of automated vulnerability fixes](#)," Google Security Engineering Technical Report, January 31, 2024; Raluca Ada Popa and Four Flynn, "[Introducing CodeMender: an AI agent for code security](#)," Google DeepMind, October 6, 2025.
21. Kamile Lukosiute, "[Design for the defenders you care about or risk being useless](#)," December 8, 2025.
22. Verizon, "[2024 Data Breach Investigations Report](#)," 2024.
23. Houjun Liu et al., "[SecureForge: Finding and Preventing Vulnerabilities in LLM-Generated Code via Prompt Optimization](#)," arXiv, May 8, 2026; CISA, "[Shifting the Balance of Cybersecurity Risk: Principles and Approaches for Secure by Design Software](#)," 2023.
24. Chun Jie Chong, Zhihao (Zephyr) Yao, and Iulian Neamtiu, "[Artificial-Intelligence Generated Code Considered Harmful: A Road Map for Secure and High-Quality Code Generation](#)," arXiv, October 12, 2024.
25. Arup Datta, Ahmed Aljohani, and Hyunsook Do, "[Secure Code Generation at Scale with Reflexion](#)," arXiv, November 5, 2025.
26. Heather Adkins and Four Flynn, "[Evaluating Threats & Automating Defense at Google](#)," [un]prompted, March 25, 2026; Jingxuan He et al., "[Instruction Tuning for Secure Code Generation](#)," arXiv, July 12, 2024.
27. Herbie Bradley and Girish Sastry, "[The Great Refactor](#)," Institute for Progress, August 11, 2025.
28. Ruth C., "[10 questions to ask when using AI models to find vulnerabilities](#)," National Cyber Security Centre, May 11, 2026.
29. Shaun Ee and Pegah Maham, "[The Three Layers of Agent Security: A Framework for Policymakers](#)," Google DeepMind, June 2026.

30. UK AISI, "[Incident Report: unsanctioned agent behaviour during cyber testing](#)," August 4, 2026.
31. Frontier Model Forum, "[Chain of Thought Monitorability](#)," January 27, 2026.
32. UK AISI, "[Security Incident INC-2026-07-28-01](#)," August 4, 2026.
33. Frontier Model Forum, "[Managing Advanced Cyber Risks in Frontier AI Frameworks](#)," February 13, 2026.
34. Lauren Deason et al., "[CyberSOCEval: Benchmarking LLMs Capabilities for Malware Analysis and Threat Intelligence Reasoning](#)," arXiv, September 15, 2025; Yiran Wu et al., "[ExCyTIn-Bench: Evaluating LLM agents on Cyber Threat Investigation](#)," arXiv, May 1, 2026; Nancy Lau et al., "[ZeroDayBench: Evaluating LLM Agents on Unseen Zero-Day Vulnerabilities for Cyberdefense](#)," *ICLR 2026 Workshop on Agents in the Wild*, March 2, 2026.
35. Joshua Saxe, "[The Hard Part Isn't Building the Agent: Measuring Effectiveness](#)," [un]prompted, March 25, 2026.